

DENODO ARACNE 4.5 GUÍA DEL ADMINISTRADOR

Update 2 (18 Dic, 2008)

C/ Alejandro Rodríguez, 32
28039 MADRID
Telf. (+34) 912 77 58 55
Fax. (+34) 912 77 58 60

C/ Real, 22 3º
15003 A CORUÑA
Telf. (+34) 981 10 02 00
Fax. (+34) 981 10 02 05

530 Lytton Avenue, Suite 302
Palo Alto, CA 94301 USA
Phone (650) 566 8833
Fax (650) 566 8836

www.denodo.com

ÍNDICE

ÍNDICE DE FIGURAS

PREFACIO

ALCANCE

Este documento presenta el sistema de crawling, indexación y búsqueda Denodo Aracne.

QUIÉN DEBERÍA USAR ESTE MANUAL

Este documento está dirigido a administradores que pretendan instalar, configurar y/o utilizar Denodo Aracne en aplicaciones de *crawling* o indexación y búsqueda de información procedente de la Web, sistemas de ficheros, servidores de correo electrónico, etc.

RESUMEN DE CONTENIDOS

Más concretamente, en este documento se describen:

- Los procedimientos de instalación del Software Denodo Aracne.
- Configuración del sistema para su posterior utilización.
- Operación del sistema utilizando su herramienta de administración Web.
- Construcción de buscadores sobre la información recolectada y extensión de las funcionalidades del sistema utilizando la API Denodo Aracne.

1 INTRODUCCIÓN

La suite de productos de Denodo Technologies proporciona funcionalidades avanzadas para la integración de información procedente de fuentes dispersas, heterogéneas y que, posiblemente, presentan un bajo nivel de estructuración.

Denodo Aracne permite el *crawling*, indexación y consulta de información no estructurada en una amplia variedad de formatos.

Entre las principales características de Denodo Aracne se encuentran:

- *Crawling* web avanzado capaz de tratar páginas web de cualquier nivel de complejidad que incluyan características como JavaScript, HTML dinámico, autenticación, redirecciones complejas, menús emergentes, etc.
- Crawling de servidores FTP y de sistemas de ficheros.
- Posibilidad de recuperar el contenido de mensajes de correo electrónico accesibles vía POP3 o IMAP.
- Crawling de cuentas de correo de Microsoft Exchange Server,
- Rápida indexación: una media de 200MB/hora.
- Pequeño tamaño de índices: aproximadamente el 30% del tamaño del texto indexado.
- Soporte para los formatos más populares: HTML, texto, XML, MS Word, RSS (versiones 0.91, 0.92, 1.0 y 2.0), PDF, MS Excel, MS PowerPoint, EML, etc.
- Búsquedas complejas: soporte para operadores AND, OR, NOT, +, -, uso de paréntesis, uso de comodines, búsquedas por frase exacta, búsquedas multicampo (título, URL, etc.).
- Mantenimiento de índices mediante la eliminación de documentos antiguos, obsoletos, no accesibles, etc.

La planificación y configuración de las tareas de crawling ejecutadas por Denodo Aracne se realiza a través del módulo Denodo Scheduler. Véase [SCHED] para información detallada al respecto.

2 ARQUITECTURA GENERAL

Denodo Aracne se divide en dos módulos independientes:

- **Aracne Server (ARN-CRAWLER)**: El módulo de crawling constituye una herramienta de recuperación automática de información no estructurada disponible en la Web, sistemas de ficheros, servidores de correo electrónico, etc. (ver Figura 1). Denodo Aracne dispone de una serie de crawlers para diferentes fuentes de información no estructurada.
- **Aracne Search/Index Engine Server (ARN-INDEXER)**: El módulo de indexación y búsqueda sobre índices permite almacenar documentos para permitir realizar posteriormente búsquedas sobre ellos.

Denodo Aracne también incluye una herramienta de administración de configuración y de gestión y búsqueda sobre índices.

La forma de utilización normal de Denodo Aracne es a través del planificador de tareas de la Plataforma Denodo, Denodo Scheduler [SCHED]. En particular, definiendo tareas de tipo ARN para cualquiera de los motores de crawling implementados por Denodo Aracne o tareas ARN-Index para operaciones de mantenimiento automático sobre los índices de ARN-Indexer, como eliminación de documentos antiguos, obsoletos, no accesibles, etc.

Por otra parte, el servidor de índices puede también utilizarse en la definición de cualquier tipo de tarea de extracción de Denodo Scheduler, para exportar las tuplas obtenidas como documentos en un índice, de tal forma que se puedan realizar posteriormente complejas búsquedas booleanas por palabra clave sobre ellos.

En la Figura 1 se muestra la arquitectura de Denodo Aracne, con sus dos servidores, de crawling e indexación/búsqueda, y su relación con Denodo Scheduler. Adicionalmente, Denodo Aracne posee su propia API de indexación/consulta (ver sección 5.1)

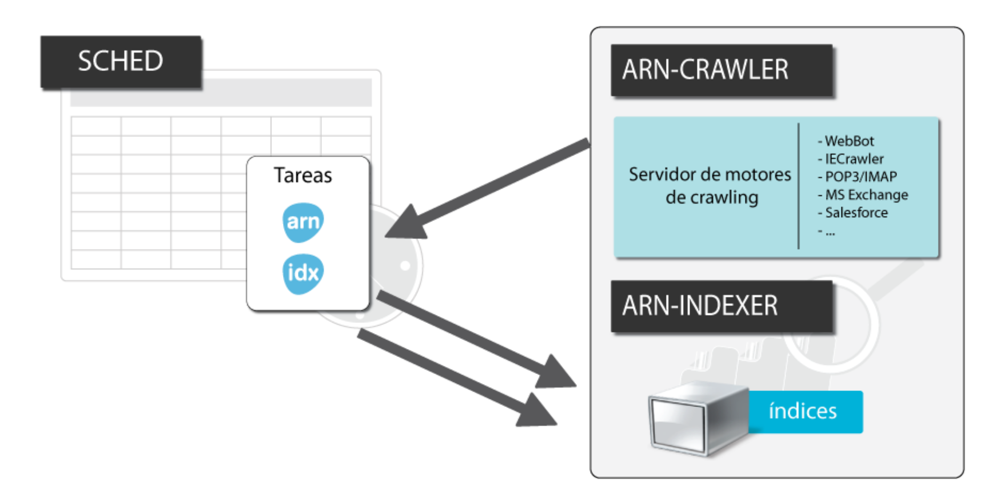


Figura 1 Arquitectura de Denodo Aracne

El núcleo de ARN-Crawler lo constituyen los robots de crawling:

- **WebBot** e **IECrawler** atraviesan la estructura de hipertexto de la Web, partiendo de un conjunto de URLs iniciales y recuperan, de forma recursiva, todas las páginas accesibles desde el conjunto de URLs de partida. Permiten además conectarse a un servidor FTP y obtener la información contenida en todos los ficheros y subdirectorios de un directorio especificado como URL inicial.
  WebBot es capaz, además, de explorar un sistema de ficheros considerando como URL inicial un directorio y extrayendo la información contenida en todos sus ficheros y subdirectorios.

- **Crawler POP3/IMAP.** Permite recuperar información de correos electrónicos contenidos en servidores accesibles a través de los protocolos POP3 o IMAP. Incluye soporte para ficheros adjuntos.
- **Crawler MS Exchange.** Permite recuperar información de correos electrónicos contenidos en servidores MS Exchange [MSEX]. Incluye soporte para ficheros adjuntos.
- **Crawler Salesforce.com.** Permite recuperar información contenida en entidades de datos accesibles a través de una cuenta en el servicio on-line Salesforce.com [SLF].
- **CustomCrawler** permite extraer la información de una fuente de datos, a través de una implementación Java proporcionada por el administrador de Denodo Aracne. Este tipo de robot permite la construcción ad-hoc de un crawler para una fuente específica.

La configuración de cada tipo de crawler concreto se describe en detalle en la Guía de Administrador de Denodo Scheduler [SCHED], que es dónde se crean las tareas de extracción ARN. Lo mismo es aplicable a las acciones de mantenimiento de ARN-Indexer.

El motor de consulta (ver Figura 1) recibe consultas de los usuarios a través de la interfaz web o de la API Aracne de búsqueda, recupera los resultados relevantes a esa consulta, utilizando la información contenida en el índice y muestra la respuesta obtenida al usuario en forma de listado de documentos.

El módulo de indexación y búsqueda permite:

- A través de Denodo Scheduler, indexar documentos en diversos formatos: HTML, PDF, Ms. Word, Excel, PowerPoint, RSS (versiones 0.91, 0.92, 1.0 y 2.0), EML, etc.
- Realizar indexaciones y búsquedas de documentos con mayor fiabilidad, al no limitarse a búsquedas de palabras exactas, sino que las asociaciones pueden ser realizadas en base al lema/raíz de las mismas.
- Representar y realizar consultas sobre las diversas partes de un documento: título, resumen, cuerpo, etc.
- Tener varios índices, lo que posibilita la creación de distintos buscadores temáticos.
- Ordenación de resultados por relevancia basada en el algoritmo TFIDF.
- Búsquedas avanzadas con operadores +, - , *, AND, OR, búsqueda por similitud de palabras (fuzzy), búsquedas por proximidad configurable de los términos, etc.

3 INSTALACIÓN Y EJECUCIÓN

La *Guía de Instalación de la Plataforma Denodo* [DENINST] proporciona toda la información necesaria para instalar Denodo Aracne, incluyendo los requisitos mínimos de hardware y software, e instrucciones para la utilización de la herramienta de instalación y para la configuración inicial del sistema.

El servidor de Denodo Aracne consta de tres procesos servidores:

- Servidor de crawling (ARN-CRAWLER). Este servidor se encarga de la ejecución de las tareas de crawling.
- Servidor de indexación/búsqueda (ARN-INDEXER). Este servidor se encarga de las labores de indexación de información en el repositorio. También es capaz de ejecutar consultas de la misma forma que el servidor de búsqueda.
- Servidor de crawling MS Exchange. El crawler de MS Exchange requiere de un servidor propio que haga de Proxy contra el servidor MS Exchange. El crawler MS Exchange se comunica con este servidor para realizar las peticiones de correos electrónicos y obtenerlos.
- Servidor de administración web. Servidor que da soporte a la herramienta de administración web de los servidores de crawling y de indexación de Denodo Aracne.

Los servidores pueden arrancarse y detenerse utilizando la herramienta Denodo Platform Control Center (ver *Guía de Instalación de la Plataforma Denodo* [DENINST]). Para conectarse a la herramienta de administración es necesario utilizar el usuario **admin**, con contraseña inicial **admin**. El URL de acceso por defecto a la herramienta de administración web desde una máquina local es `http://localhost:9090/webadmin/denodo-aracne-admin`.

Como alternativa se proporcionan scripts en la ruta `$DENODO_HOME/bin`. Para cada servidor existe un script `servername_startup.sh` (`servername_startup.bat` y `servername_startup.exe` en Windows) para arrancarlo y un script `servername_shutdown.sh` (`servername_shutdown.bat` y `servername_shutdown.exe` en Windows) para detenerlo. Por ejemplo, para el servidor ARN-CRAWLER los scripts reciben el nombre `arn_startup.sh` y `arn_shutdown.sh`. Para arrancar y detener la herramienta de administración web existen los scripts `arn_webadmin_startup.sh` y `arn_webadmin_shutdown.sh` respectivamente (`arn_webadmin_startup.bat` y `arn_webadmin_shutdown.bat` en Windows).

En el caso de máquinas Windows, se incluyen scripts para instalar los servidores como servicio. Los scripts reciben el nombre `servernameservice.bat` (e.g. `arnservice.bat`).

4 ADMINISTRACIÓN

La *Guía de Instalación de la Plataforma Denodo* [DENINST] proporciona información detallada sobre las tareas de configuración que es necesario realizar antes de ejecutar Aracne.

En la siguiente sección se describen las opciones de configuración del servidor y los *logs* del sistema.

4.1 AUTENTICACIÓN

Al acceder a la herramienta de administración de Denodo Aracne se muestra una pantalla inicial de autenticación (ver Figura 2), en la que el usuario deberá introducir la contraseña de administración. El usuario también debe especificar a qué servidor desea conectarse: al servidor de crawling (Aracne Server) o al servidor de indexación/búsqueda (Aracne Search/Index Engine Server). También proporciona la posibilidad de recordar la contraseña para futuras autenticaciones.

En la misma pantalla se permite la opción de modificar el servidor Denodo Aracne (nombre del servidor y puerto del proceso de administración) contra el que se conectará la herramienta.

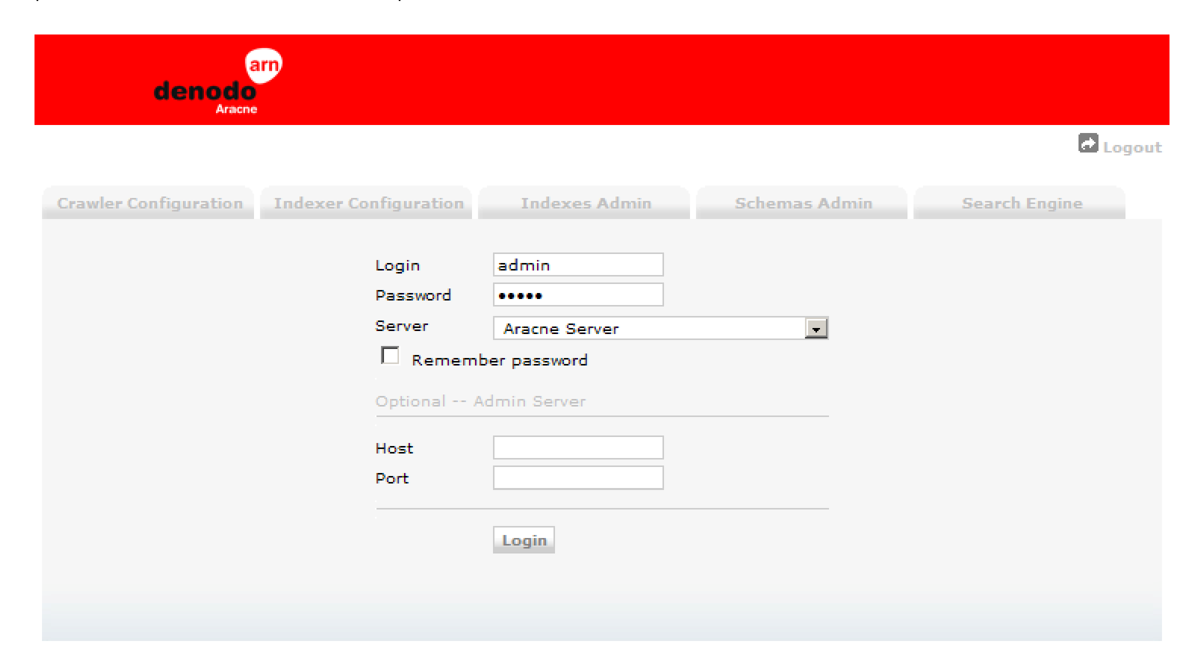


Figura 2 Pantalla de autenticación

4.2 ADMINISTRACIÓN DEL SERVIDOR ARACNE (ARN-CRAWLER)

Una vez autenticado en la herramienta de administración del servidor de crawling, el usuario puede acceder a la pestaña de configuración, dónde se le presentan las siguientes posibilidades (ver Figura 3):

- Modificar la contraseña de administración.
- Cambiar la configuración de acceso a través de Proxy del *crawler* WebBot y del servidor Aracne (ver sección 4.2.1).
- Indicar otro servidor de administración Aracne al que conectarse. Para ello debe utilizarse la opción *Change Current Admin Server* e indicar el nombre de máquina y el puerto de ejecución del servidor que se desea pasar a administrar.

- Cambiar los puertos utilizados por el servidor (opción *Configure Server Ports*). La modificación tendrá validez la próxima vez que se arranque el servidor.
- Configurar el nivel de concurrencia del servidor (ver sección 4.2.2).

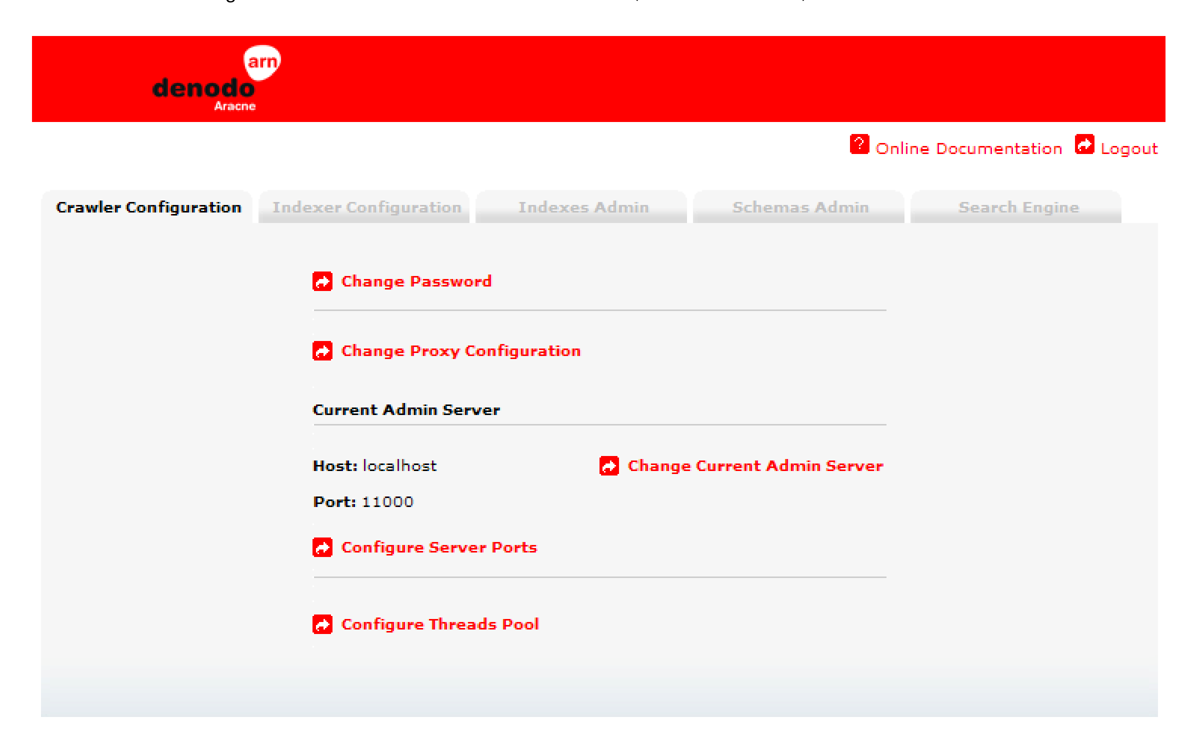


Figura 3 Pantalla de configuración general

4.2.1 Configuración de Proxy

Cuando los crawlers web deban acceder a la información a través de un proxy, será necesario configurar la información del mismo.

Desde la pestaña de edición de configuración de la herramienta de administración es posible configurar un Proxy para el crawler WebBot y el servidor Aracne. Para ello debe pulsarse la opción *Change Proxy Configuration* y especificar los siguientes parámetros (ver Figura 4):

- **Host**. Nombre o dirección IP de la máquina que actúa como Proxy.
- **Port**. Número de puerto en el que está lanzado el servicio de Proxy.
- **Login**. Identificador de usuario en el Proxy.
- **Password**. Contraseña del usuario en el Proxy.
- **Realm/NTLM Domain**. Dominio de seguridad del Proxy. Normalmente puede verse el valor esperado en el cuadro de diálogo de autenticación mostrado por el proxy.

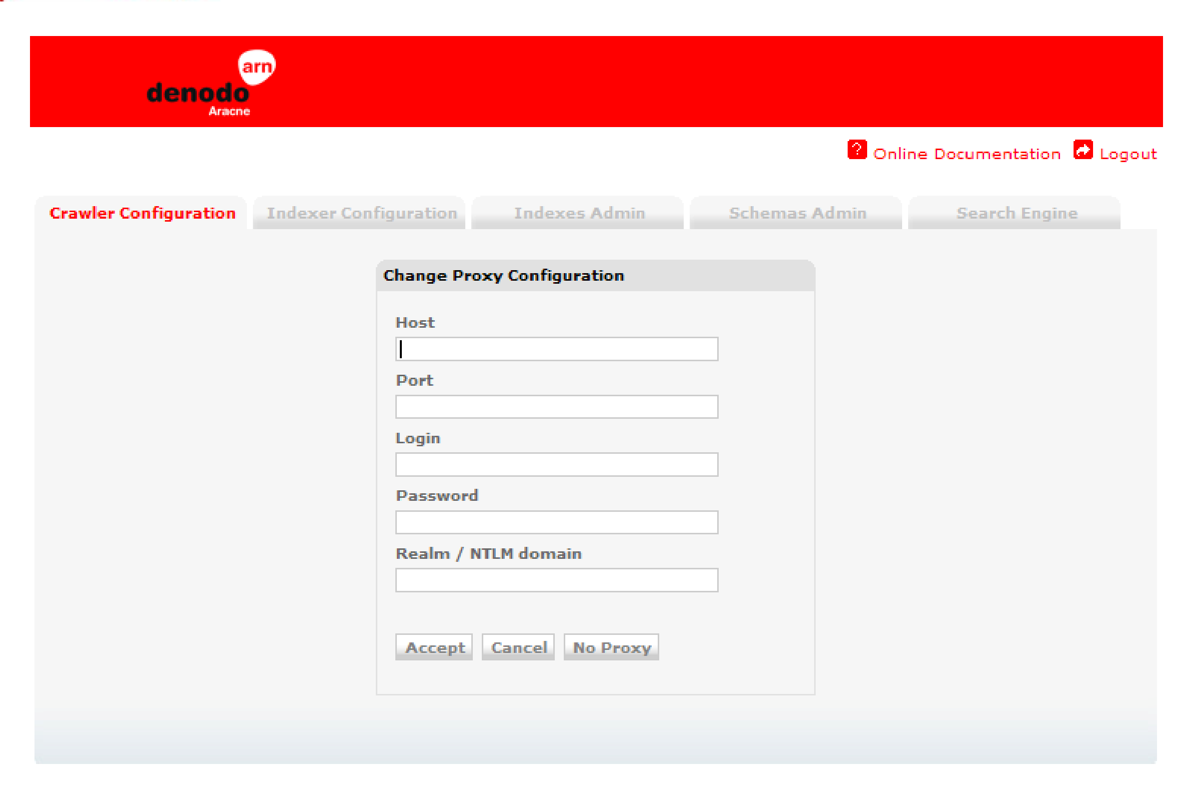


Figura 4 Configuración de proxy

En IECrawler, para un acceso vía proxy es necesario configurar los *browsers* de IECrawler para que utilicen *Proxy* (la configuración se realiza de la misma forma que en Microsoft Internet Explorer).

4.2.2 Hilos de Ejecución

El servidor Aracne utiliza un pool de threads reutilizables para gestionar la ejecución de las múltiples consultas que puede generar una misma tarea. Los parámetros que es posible configurar son los siguientes:

- **Normal number of threads**. Representa el número de threads en el pool a partir del cual se reutilizan los threads inactivos (por defecto 20). Mientras en el pool haya menos de este número de threads, se seguirán creando nuevos threads. Cuando se solicite un thread, y el número de threads en el pool iguale o supere este valor, se devolverán threads inactivos si existen; en caso contrario se seguirán creando nuevos threads hasta llegar al valor establecido por el siguiente parámetro. Intuitivamente, este parámetro indica el número de threads que el sistema debería de tener activos simultáneamente en condiciones normales de carga.
- **Maximum number of threads**. Representa el número máximo de threads del pool (por defecto 60).
- **Keep alive time (ms).** Especifica el tiempo máximo en milisegundos que un thread inactivo permanece en el pool, si el número de threads totales supera el indicado en Normal number of threads (por defecto 0). Si el valor es 0, entonces los threads creados por encima de este valor, una vez terminada la ejecución de su tarea, finalizan. En caso contrario, finalizan aquellos que excedan el tiempo especificado en este parámetro.

4.3 ADMINISTRACIÓN DEL SERVIDOR DE BÚSQUEDA/INDEXACIÓN (ARN-INDEXER)

Una vez autenticado en la herramienta de administración del servidor de indexación/búsqueda, el usuario puede acceder a una de las siguientes funcionalidades:

- Configuración del servidor de indexación (ver sección 4.3.1).
- Administración de índices (ver sección 4.3.2).
- Administración de esquemas de índices (ver sección 4.3.3)
- Motor de búsqueda (ver sección 4.3.4).

4.3.1 Configuración del servidor

La pestaña de configuración del servidor permite las siguientes posibilidades:

- Modificar la contraseña de administración.
- Indicar otro servidor de administración Aracne al que conectarse. Para ello debe utilizarse la opción *Change Current Admin Server* e indicar el nombre de máquina y el puerto de ejecución del servidor que se desea pasar a administrar.
- Cambiar los puertos utilizados por el servidor (opción *Configure Server Ports*). La modificación tendrá validez la próxima vez que se arranque el servidor.
- Exportar la metainformación de índices y esquemas del servidor (opción *Export*). Esta funcionalidad es especialmente útil para propósitos de respaldo ("backup") y/o migración a otras instalaciones de ARN-Indexer. Para ello se genera un fichero comprimido con Zip, conteniendo toda la información necesaria para restablecer la metainformación del servidor al estado de ese momento (incluye también el fichero de configuración del servidor). La plataforma proporciona scripts para este mismo propósito (ver apéndice 6.2).
- Importar la configuración, índices y esquemas de índices a partir de un fichero que contiene el estado de un servidor en un determinado momento (opción *Import*). Es posible especificar si se desea reemplazar elementos existentes por los incluidos en el fichero que se está importando, en el caso de que ya exista un índice o esquema de índices con el mismo nombre. Esta funcionalidad es especialmente útil para propósitos de migración. ARN-Indexer incluye scripts para este mismo propósito (ver apéndice 6.2).

4.3.2 Administración de Índices

El servidor de indexación gestiona un conjunto de índices en los que se pueden almacenar documentos o sobre los que se pueden realizar consultas.

La pantalla de administración de índices permite crear nuevos índices, editar la configuración de los índices existentes, o borrarlos. Para crear o editar un índice, es necesario especificar la siguiente información:

- **Index name**. El nombre del índice.
- **Index path**. La ruta en el sistema de ficheros en la que se almacenará físicamente la metainformación y datos del índice.
- **Analyzer type**. El tipo de analizador especifica qué tokens de un texto son considerados en el momento de resolver una consulta. El analizador a utilizar debe escogerse en función del idioma esperado para los documentos a indexar y de si se desea o no aplicar técnicas de "stemming" y eliminación de tokens muy habituales (lista de palabras de parada).

  Los analizadores que utilizan "stemming" tratan de eliminar las terminaciones morfológicas más comunes de las palabras de un documento, antes de que éste sea indexado.

  El objetivo es conseguir que una búsqueda por una determinada palabra clave devuelva también los documentos que contienen otras palabras con la misma raíz léxica. Por ejemplo, si se busca la palabra "comercio", se devolverán también los documentos que contengan palabras tales como "comerciar", "comercios" o "comerciando".

  Dependiendo del uso que vaya a recibir la aplicación, las técnicas de stemming pueden ser convenientes o no. También es necesario tener en cuenta que las técnicas de stemming se basan en

una serie de reglas generales que pueden admitir ciertas excepciones. Esto quiere decir que en algunos casos raros, el sistema puede identificar erróneamente las raíces léxicas de algunas palabras.

Denodo Aracne incluye tres analizadores diferentes.

- **standard**. Considera lista de palabras de parada en inglés pero no usa stemming.
- **english**. Considera lista de palabras de parada y stemming para el idioma inglés.
- **spanish**. Considera lista de palabras de parada y stemming para el idioma inglés.

- **Schema**. El esquema del índice permite especificar qué campos serán incluidos en el índice y con qué propiedades. En la sección 4.3.3 se describe cómo administrar esquemas de índices y la configuración del esquema incluido por defecto con Denodo Aracne (standard).

La pantalla de administración de índices también permite borrar el contenido de un índice (enlace **Delete Index Content**).

NOTA: La distribución de Denodo Aracne incluye pre-creado el índice "default", que utiliza el analizador "Standard" y el esquema "Standard".

4.3.3 Administración de Esquemas de Índices

Denodo Aracne permite configurar qué campos va a tener un índice y las características de indexación de los mismos. Para ello es posible crear diferentes esquemas, que serán utilizados en la configuración de los índices. Aunque habitualmente no es necesario, a través de la herramienta de administración de Aracne es posible crear, editar o eliminar configuraciones de esquemas de índices.

Para crear un nuevo esquema de índices es necesario especificar los siguientes parámetros:

- **Schema name**. El nombre del esquema, que será utilizado para referenciarlo desde las pantallas de creación/edición de índices.
- **Unique key**. Indica el nombre del campo del esquema que representa la clave primaria.
- **Default search field**. Especifica el nombre del campo del esquema por el que se realizarán las búsquedas sobre el índice, cuando no se especifica un campo en la consulta de forma explícita.

De forma adicional, es posible especificar información específica para determinados campos del esquema. Para especificar propiedades de indexación específicas para un campo, es necesario añadir una entrada en la sección **Customized fields**, permitiendo configurar las siguientes propiedades:

- **Name**. Nombre del campo al que se aplican las propiedades especificadas a continuación.
- **Index**. Permite especificar si se pueden realizar búsquedas por un campo y su tipo. Los valores posibles son:
  - NO. No indexar el campo, es decir, no permitir búsquedas por este campo.
  - TOKENIZED. Indexa el valor del campo en base al analizador especificado en el índice, de modo que puedan realizar búsquedas por él.
  - UN_TOKENIZED. Indexa el valor del campo, pero sin utilizar el analizador. Permite igualmente realizar búsquedas por ese campo.
- **Store**. Permite especificar si el campo se almacena en el índice. Por defecto, todos los campos se almacenan en el índice, pero opcionalmente puede especificarse que alguno no se almacene o se almacene comprimido. Los valores posibles son:
  - COMPRESS. Almacena el valor del campo en el índice en un formato comprimido. NO. No almacena el valor del campo en el índice.
  - YES. Almacena el valor original del campo en el índice.
- **Boost**. Especifica la relevancia por defecto del campo en las búsquedas. Es un valor positivo. Al subir este valor se dará más importancia en las búsquedas a los documentos que contengan ocurrencias de las palabras buscadas en este campo.
- **Search**. Especifica si se desea que el contenido del campo se almacene en el **Default search field**, para que sea incluido en las búsquedas globales (cuando no se especifica ningún campo del esquema).

En general todos los campos no binarios del documento enviado al servidor de indexación serán almacenados en el índice (**Store**="YES"), dividido en palabras (**Index**=TOKENIZED), con relevancia 1 (**Boost**=1) y su contenido se incluye en el campo **Default search field**.

El esquema de índices por defecto ("standard") define como clave primaria el campo "identifier", y como campo de búsqueda por defecto "searchableContent". También define los campos "path" y "mimetype" para que no se tokenizen (**Index**="UN_TOKENIZED") y se almacenen en el índice (**Store**="YES"). Todos los campos se incluyen en el campo de búsqueda por defecto.

4.3.4 Motor de Búsqueda

La herramienta de administración de Denodo Aracne proporciona la posibilidad de realizar búsquedas y operaciones de mantenimiento sobre los índices creados. Para ello es necesario seguir el enlace _Search Engine_ de la pantalla inicial de la herramienta de administración del servidor de índices.

En la pantalla del motor de búsqueda (ver Figura 5) se proporcionan dos posibilidades de búsqueda sobre el índice:

- Búsqueda por palabra clave. Es necesario introducir los siguientes parámetros:
    - **Keywords**. Las palabras clave necesarias para la búsqueda (en el apéndice 6.1 puede consultarse la sintaxis de búsqueda).
    - **Results per page**. Indica el número de resultados que se desea por página. El valor por defecto es 10.
    - **Index**. Especifica el nombre del índice sobre el que se desee realizar la búsqueda. En este caso será *default*.
- Obtención de la lista de todos los documentos del índice
    - En esta modalidad de búsqueda sólo es necesario modificar, si se desea, el número de resultados a mostrar por página (**Results per page**), que tomará como valor por defecto 10.
    - Además es necesario indicar el nombre del índice (**Index**) sobre el que realizar la búsqueda.

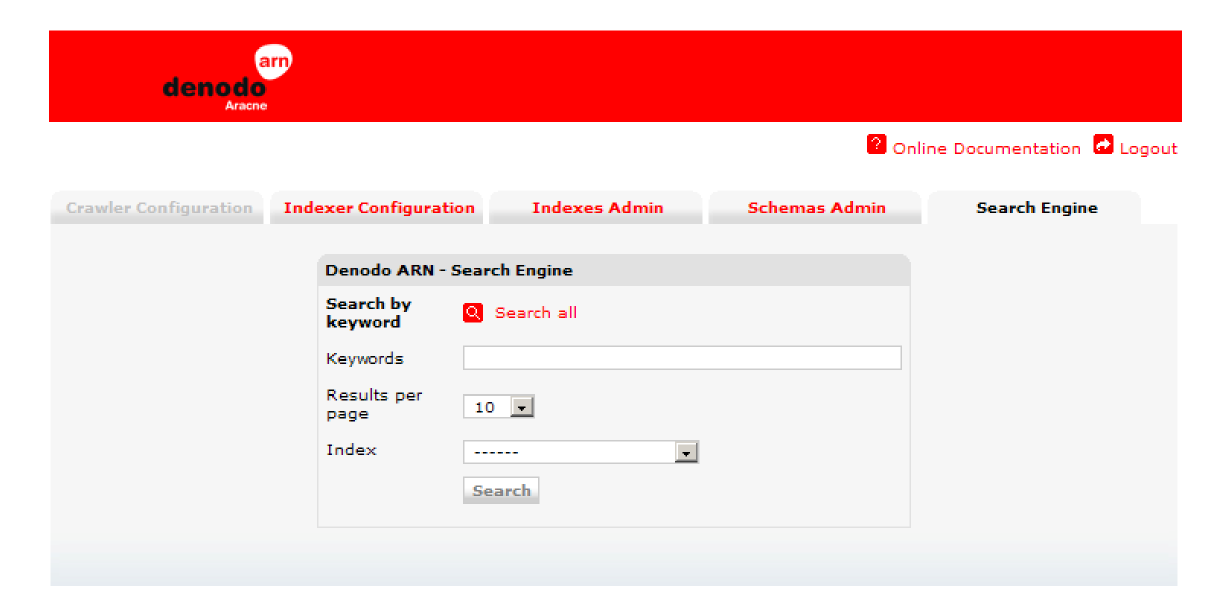


Figura 5 Pantalla del motor de búsqueda

Una vez realizada una consulta de cualquiera de los tipos anteriores, se muestra el resultado de la búsqueda (ver Figura 6).

Si el número de resultados es mayor que el número especificado en la búsqueda, los resultados aparecen paginados, con lo que es necesario navegar por los enlaces de paginación (*previous* y *next*) para examinar todos los resultados obtenidos.

Para cada resultado se muestran sus campos, que dependen del tipo de documentos indexados y del esquema de índice considerado.

denodo arn Aracne

Online Documentation Logout

Crawler Configuration | Indexer Configuration | Indexes Admin | Schemas Admin | Search Engine

Denodo ARN - Search Engine

Search results 25 document/s found   Search by keyword   Search all

Index: default   Remove selected documents [Remove]

Select: All - None   Remove all documents matching the submitted query [Remove All]

1
URL: http://www.denodo.com/english/copyright.html
PATH: denodo_4/www.denodo.com/YExYUOEnTmegwvLXnkW5zg==.html
TITLE: Denodo Technologies
ANCHORTEXT: Copyright
MIMETYPE: text/html
_$JOB: 4
_$JOB_PROJECT: default
_$JOB_NAME: denodo
_$JOB_START_TIME: 1225294150532
_$JOB_RETRY_START_TIME: 2008-10-29 16:29:10
_$JOB_RETRY_COUNT: 0
IDENTIFIER: http://www.denodo.com/english/copyright.html
CONTENT:

Denodo Technologies is a registered trademark in USA and other countries. All registered trademarks are properties of their respective owners. Denodo Technologies S.L. relinquishes any interest in trademark property and names of others. Copyright © 2007 Denodo Technologies S.L.

SUMMARY:

Denodo Technologies is a registered trademark in USA and other countries. All registered trademarks are properties of their respective owners. Denodo Technologies S.L. relinquishes any interest in trademark property and names of others. Copyright © 2007 Denodo Technologies S.L.

_$URIERRORS: 0
INDEXSCORE: 0.29722592

2
URL: http://www.denodo.com/english/resources/index3.php
PATH: denodo_4/www.denodo.com/UMw3wQAYvNTHl7bofnPnHQ==.php
TITLE: Denodo Technologies
ANCHORTEXT: Web Automation
MIMETYPE: text/html
_$JOB: 4
_$JOB_PROJECT: default
_$JOB_NAME: denodo
_$JOB_START_TIME: 1225294150532
_$JOB_RETRY_START_TIME: 2008-10-29 16:29:10
_$JOB_RETRY_COUNT: 0
IDENTIFIER: http://www.denodo.com/english/resources/index3.php
_$URIERRORS: 0
INDEXSCORE: 0.2547651

Figura 6 Resultados de búsqueda para la consulta: denodo

Desde la pantalla de resultados de la búsqueda es posible eliminar cualquier documento del índice. Para ello cada documento del resultado aparece asociado a un *checkbox* y al seleccionar uno o varios documentos y pulsar en el botón "*Remove*" estos resultados serán eliminados del índice. Existe, además, la posibilidad de eliminar todos los documentos resultado de una búsqueda sobre el índice pinchando en el botón "*Remove All*".

4.4 CONFIGURACIÓN DE LOGS

Denodo Aracne posee en la ruta `DENODO_HOME/conf/arn` (donde `DENODO_HOME` se refiere a la ruta base de instalación) el fichero de configuración de log del servidor de crawling y en la ruta `DENODO_HOME/conf/arn-index` el fichero de configuración del servidor de indexación/búsqueda. Estos ficheros están basados en Log4j [LOG4J]. Entre otras posibilidades, se permite modificar la ruta donde se almacenan los ficheros de *log* y el nivel de *log* de las categorías definidas en la aplicación. Para más información, véase la documentación de Log4j. El servidor de crawling genera un fichero de nombre `arn.log` en la ruta `DENODO_HOME/logs/arn` y el servidor de indexación otro de nombre `arn-index.log` en la ruta `DENODO_HOME/logs/arn-index`.

La herramienta de administración web también posee un fichero de configuración `log4j.xml` para establecer el nivel de registro de los eventos generados por esta aplicación. Este fichero se encuentra en el directorio `DENODO_HOME/resources/apache-tomcat/webapps/webadmin/denodo-aracne-admin/WEB-INF/classes`. La herramienta de administración genera dos ficheros de log:

- `DENODO_HOME/logs/arn/arn-admin.log`. Contiene información de ejecución de la herramienta de administración.
- `DENODO_HOME/logs/apache-tomcat/denodo-tomcat.log`. Contiene información relacionada con el arranque/instalación/parada de la herramienta de administración en el servidor web.

La configuración de *log* de los procesos de *crawling* realizados con IECrawler se encuentra en `DENODO_HOME/conf/arn/iecrawler`. Se crean los siguientes ficheros de log en la ruta `DENODO_HOME/logs/arn/iecrawler` (cada tipo de log almacena hasta 10 ficheros de respaldo como máximo, con un tamaño de 10MB cada uno):

- *`nombre_tarea`*`.log`: Fichero que contiene el flujo de eventos del *crawling*. Es posible especificar un nombre de fichero, en lugar de un directorio, para el elemento `ROLLINGFILE` *`filedefault`* en el fichero de configuración de *logs* `log.xml.` En este caso un único fichero de *log* contendría el flujo de eventos de todas las tareas IECrawler que se ejecuten en el sistema, en lugar de un fichero por tarea (configuración por defecto).
- `access_url.log`: Contiene el listado de URLs a los que ha accedido el *crawler*.
- `accept_url.log`: Contiene el listado de URLs que el *crawler* ha aceptado para procesar.
- `reject_url.log`: Contiene el listado de URLs que han sido descartados por el *crawler*, indicando el motivo.
- `error_url.log`: Contiene el listado de URLs que han producido un error al acceder a ellos (por ejemplo errores HTTP 404 que no estén capturados por el servidor).

La configuración de *log* de los procesos de *crawling* realizados con ExchangeMailCrawler se encuentra en `DENODO_HOME/conf/arn/exchangecrawler`. El *crawling* genera un fichero de *log* en la ruta `DENODO_HOME/logs/arn/exchangecrawler` (almacena hasta 10 ficheros de respaldo como máximo, con un tamaño de 10MB cada uno):

- `exchangemailcrawler.log`: Fichero que contiene el flujo de eventos del *crawling*.

5 API DENODO-ARACNE

5.1 API CLIENTE – BÚSQUEDA / INDEXACIÓN

Además de proporcionar una interfaz Web ya construida para realizar búsquedas sobre la información descargada e indexada, Denodo Aracne también permite implementar un buscador propio acorde a necesidades más concretas.

Para ello la plataforma posee una fachada `com.denodo.arn.index.client.IndexManager`, que permite las siguientes funcionalidades:

- `find`: *obtiene* los documentos del índice, paginando en base a un índice de inicio y un número de documentos. Este método presenta diversas firmas: si se le pasa una consulta, devuelve sólo los documentos obtenidos en base a esa "query"; en caso contrario, devuelve todos los documentos del índice. También permite activar el la funcionalidad de *highlighting*, de modo que las palabras de la consulta aparecen resaltadas (con un color diferente) en los resultados de la búsqueda (se puede utilizar la configuración por defecto o especificarla en el objeto `HighlightConfig)`. Además, para el caso de documentos que provengan de extracciones contra VDP, se puede utilizar el objeto `MainTermsConfig` que crea un nuevo campo con los datos más relevantes de los campos del documento.
- `getFields`: obtiene una lista de todos los campos únicos que existen en el índice.
- `listIndices`: obtiene la lista de manejadores de resultados de Aracne que permiten la realización de búsquedas. Cada uno de estos manejadores, `IndexEngineMetadata`, incluye la lista de campos del esquema de índice que representan. Adicionalmente, pueden incluir una lista de campos generados tras una búsqueda, como por ejemplo la relevancia de un documento en una búsqueda realizada.
- `addDocument`: añade un documento a un índice.
- `addDocuments`: añade documentos a un índice.
- `addElement`: crea un nuevo elemento usando la configuración que recibe como parámetro.
- `changePassword`: permite a los clientes cambiar su password.
- `changeServerPorts`: permite modificar los números de puerto usados por el servidor.
- `createIndex`: permite crear un nuevo índice.
- `delete`: permite borrar un índice y todos los documentos que contiene.
- `deleteDocuments`: permite borrar (de uno o varios índices) todos los documentos obtenidos como resultado de una consulta.
- `deleteDocuments`: elimina documentos de un índice en base a una consulta o a un valor para un campo específico del índice.

Los métodos de `find` devuelven un objeto de tipo `DocumentChunk`. Esta clase representa una colección de objetos de la clase `Document` que encapsula a cada documento devuelto por una consulta sobre un manejador de índice. La clase `Document` constituye la representación interna de un documento del índice con una serie de campos variables (entradas de un mapa). En caso de realizar una búsqueda en base a una expresión de consulta, la clase `Document` también incluye el campo: resumen destacado. Este campo se puede mostrar entre marcas especiales, para lo cual se debe activar el highlight.

Los métodos de búsqueda `find` pueden recibir los siguientes parámetros:

- `index`, indica el `nombre` del manejador de índice sobre el que realizar la búsqueda. Identifica el nombre del índice, el analizador y las *extensiones del esquema de índice* utilizadas para la creación del mismo.
- `startIndex`, indica el número del primer resultado que se quiere obtener.
- `count`, indica el número de resultados que se desea obtener con la consulta.

- `query`, indica la consulta que ha introducido el usuario.
- `mainTermsConfiguration` indica la configuración necesaria para obtener los términos más relevantes de los campos del índice deseados (ver sección 5.1.1).
- `enableHighlight`, permite activar la aparición de palabras entre marcas en los resultados de una consulta.
- `highlightConfig` permite especificar la configuración de las marcas para las palabras resaltadas.

Para más información consultar la documentación Javadoc de Denodo Aracne y los ejemplos en `DENODO_HOME/samples/arn/arn-index-api`.

5.1.1 Términos más relevantes de un documento

Denodo Aracne es capaz de generar automáticamente las palabras más relevantes de un documento o de un campo del mismo, de acuerdo a la medida de relevancia TFIDF (Term Frequency Inverse Document Frequency). Estos términos pueden ser obtenidos como parte del resultado de una búsqueda efectuada sobre un índice Aracne.

En el proceso de búsqueda, es posible especificar de qué campos se desea obtener los términos más relevantes utilizando una instancia de la clase `com.denodo.arn.index.client.MainTermsConfig`. Este objeto contendrá una instancia de la clase `MainTermsFieldConfig` para cada campo para el que se desee obtener términos relevantes, especificando:

- Número máximo de `términos` relevantes del campo que se incluirán para cada documento resultado de la búsqueda.
- Lista de términos relevantes a descartar (opcional). Lista de "palabras usuales" (separadas por comas) que *no* deben aparecer entre los términos más relevantes de este campo. Si Aracne generase entre los términos más relevantes del contenido del campo alguno que apareciese en dicha lista, sería eliminado de la lista de términos relevantes. Es importante darse cuenta de que aquí es necesario especificar solamente palabras usuales específicas de la aplicación. Las palabras usuales del lenguaje utilizado tales como artículos, pronombres, etc. (comúnmente llamadas "stopwords") son ya eliminadas por Denodo Aracne.

Además, la clase `MainTermsConfig` permite especificar también una lista de palabras usuales comunes a todos los campos del índice para los que se deseen obtener los términos más relevantes. Nuevamente, no es necesario preocuparse de especificar palabras usuales del lenguaje utilizado tales como artículos, pronombres, etc. (comúnmente llamadas "stopwords").

El objeto que representa cada uno de los resultados de la búsqueda, `com.denodo.commons.Document`, proporciona métodos que permiten obtener la lista de términos relevantes para cada campo del documento, como objetos `MainTerms`, ya que en el campo MAINTERMS del `Document` se almacenarían los objetos `MainTerms.`

5.2 EXTENSIONES

5.2.1 Creación de Nuevas Funciones para Expresiones Regulares

Para dotar de más potencia al lenguaje de expresiones regulares utilizado en los URLs iniciales, los filtros de enlaces y de reescritura, en los módulos de *crawling* WebBot e IECrawler es posible crear nuevas funciones al estilo de *DateFormat* (ver Guía del Administrador de Scheduler [SCHED]).

Para crear nuevas funciones en WebBot, implementado en Java, es necesario extender la clase:

```
com.denodo.aracne.webbot.util.processors.function.FunctionExpression
```

implementando sus dos métodos abstractos:

```
public abstract void validate(),
```

que comprueba que los parámetros de la función son válidos y, en otro caso, lanza una `ExpressionProcessorException`.

y

```
public abstract String doProcess(),
```

que evalúa la función y la sustituye por su valor correspondiente. En el caso de la función `DateFormat`, `doProcess` calcula la fecha que se corresponde con una expresión determinada y la devuelve como un String siguiendo el patrón indicado.

La clase `FunctionExpression` proporciona dos métodos utilidad:

```
public String getParameter(int i) throws InvalidParameterException
public int getNumOfParameters()
```

que permiten obtener los parámetros de la función y su número para facilitar la implementación de nuevas funciones.

Para que las nuevas funciones puedan ser utilizadas por Denodo Aracne, éstas deben pertenecer al paquete `com.denodo.aracne.webbot.util.processors.function`.

Para más información consultar la documentación Javadoc de Denodo Aracne y los ejemplos en `DENODO_HOME/samples/aracne/webbot-api`.

Para crear una nueva función en IECrawler, implementado en C++, es necesario generar una nueva DLL con una clase, por ejemplo `NewFunction`, que extienda la clase `CFunctionExpression` y que implemente el siguiente método:

```
CProcessorResult* DoProcess();
```

que evalúa la función y devuelve su valor en forma de `CProcessorResult`.

Además la clase `CFunctionExpression` proporciona dos métodos utilidad:

```
int GetNumOfParameters();
CExpression *GetParameter(int i);
```

que permiten obtener los parámetros de la función en forma de `CExpression` y su número para facilitar la implementación de nuevas funciones.

La nueva clase debe exportar las siguientes funciones:

```
extern "C" __declspec(dllexport) void GetPlugin(CFunctionExpression**
ppFunctionExpression)
{
      *ppFunctionExpression = new NewFunction();
}

extern "C" __declspec(dllexport) void FreePlugin(CFunctionExpression**
ppFunctionExpression)
{
      delete *ppFunctionExpression;
}
```

que permiten acceder a la nueva clase y liberarla desde el IECrawler

La DLL creada debe seguir la siguiente convención de nombrado: "`nombre de la función`" seguido del sufijo "`FunctionExpression`". Continuando con el ejemplo de `NewFunction`, la DLL debería llamarse `NewFunctionFunctionExpression.dll`.

Para que las nuevas DLLs puedan ser utilizadas por Denodo Aracne deben añadirse al directorio `DENODO_HOME\dll\aracne`, donde `DENODO_HOME` denota el directorio raíz de instalación de Aracne.

Para más información sobre la creación de nuevas funciones, su compilación y vinculación consultar el fichero `README` en `DENODO_HOME/samples/aracne/iecrawler-api`, el proyecto de ejemplo en `DENODO_HOME/samples/aracne/iecrawler-api/Project` y las declaraciones de los tipos necesarios para construir nuevas implementaciones de funciones en `DENODO_HOME/samples/aracne/iecrawler-api/Libs/Include`.

NOTA IMPORTANTE: Para que Aracne funcione correctamente si se crea una nueva función para el módulo IECrawler, debe implementarse la misma función, con el mismo nombre y los mismos parámetros, para el módulo WebBot.

6 APÉNDICES

6.1 SINTAXIS DE BÚSQUEDA DE APACHE LUCENE

Lucene [LUCE] además de posibilitar la creación de consultas a través de su API, proporciona un lenguaje de consultas a través del QueryParser.

Este apéndice proporciona la sintaxis del QueryParser de Lucene, un analizador léxico que traduce una cadena de caracteres a una Query (representación interna de una consulta en Lucene) usando JavaCC.

6.1.1 Términos

Una consulta se compone de términos y operadores. Existen dos tipos de términos: términos individuales y frases.

Un término individual es una única palabra como "`hola`" o "`prueba`".

Una frase es un grupo de palabras entre comillas dobles como "`hola mundo`".

Los términos pueden combinarse entre sí mediante el uso de operadores Booleanos para formar consultas complejas (véase más abajo).

Nota: El analizador utilizado en la creación del índice será el que se utilice sobre los términos y frases de la consulta (ver apartado 4.3.2 para la configuración del analizador).

6.1.2 Campos

Lucene soporta búsquedas sobre los distintos campos de un índice. Al realizar una búsqueda se puede especificar un campo concreto o usar el campo por defecto. Los nombres de los campos y el campo por defecto es dependiente de la implementación utilizada.

Para buscar sobre un campo determinado es necesario especificar el nombre del campo seguido de dos puntos "`:`" y el término que se desea buscar.

Por ejemplo, asumiendo que un índice Lucene contiene dos campos, `título` y `texto` y `texto` es el campo por defecto, si se desea encontrar un documento titulado "`El Proyecto Jakarta`" que contiene el texto "`lucene`", entonces se puede escribir:

```
titulo:"El Proyecto Jakarta" AND texto:lucene
```

o

```
titulo:"El Proyecto Jakarta" AND lucene
```

No es necesario indicar el campo, ya que `texto` es el campo por defecto.

Nota: El campo afecta únicamente al término que aparece a continuación, por lo tanto la consulta

```
titulo:jakarta lucene
```

únicamente encontrará "`jakarta`" en el campo `título`. Encontrará "`lucene`" en el campo por defecto (en este caso el campo `texto`).

6.1.3 Modificadores de términos

Lucene admite el uso de modificadores en los términos de una consulta de manera que permite un amplio rango de opciones de búsqueda.

- **Comodines de búsqueda**

Lucene permite el empleo de caracteres comodines en los términos de búsqueda.

El símbolo '`?`' sustituye el `?` por un único carácter en la palabra. Por ejemplo, si se desea buscar "`pato`" o "`palo`" se introduciría el siguiente término:

```
Pa?o
```

El símbolo '`*`' sustituye el `*` por 0 o más caracteres. Por ejemplo, si se desea buscar "`información`" o "`informática`", se introduciría el siguiente término:

```
inform*
```

Este último comodín puede aparecer también en el medio de término.

```
te*t
```

Nota: No está permitido usar los símbolos '`*`' y '`?`' como primer carácter de una búsqueda.

- **Búsquedas difusas (Fuzzy Searches)**

Lucene permite búsquedas difusas basadas en la Distancia Levenshtein, o algoritmo de Distancia de Edición. Para realizar búsquedas difusas es necesario usar el símbolo '~' al final de un término simple. Por ejemplo, para buscar términos que se escriban de forma similar a "`votar`" se usaría la siguiente búsqueda difusa:

```
votar~
```

Esta búsqueda encontraría términos como "botar".

Se puede añadir un parámetro (opcional) que especifique la similitud requerida. Es un valor entre 0 y 1, con valores cercanos a 1 sólo los términos con un alto grado de similitud serán recuperados. Por ejemplo:

```
votar~0.8
```

Si el parámetro no se especifica su valor por defecto es 0.5.

- **Búsquedas por proximidad**

Lucene permite buscar términos entre los que haya cierta cercanía espacial. Para realizarla se utiliza el símbolo '~' al final de una Frase. Por ejemplo, para buscar "`apache`" y "`jakarta`" con una distancia de hasta 10 palabras en el mismo documento se utilizaría la búsqueda:

```
"jakarta apache"~10
```

- **Búsquedas por rango**

Las búsquedas por rango permiten recuperar documentos cuyo/s campo/s se encuentren entre un rango específico de valores. El rango especificado puede incluir los límites inferior y superior o no. La clasificación se lleva a cabo siguiendo el orden lexicográfico.

```
mod_date:[20020101 TO 20030101]
```

Esta consulta encuentra los documentos cuyo campo `mod_date` posee valores entre `20020101` y `20030101`, inclusive. La búsqueda por rango no está limitada a los campos que contengan fechas como valor:

```
titulo:{Aida TO Carmen}
```

Esta consulta recupera todos los documentos cuyos títulos se encuentren entre "`Aida`" y "`Carmen`", no inclusive.

Los rangos inclusivos se especifican mediante corchetes y los exclusivos mediante llaves.

- **Aumento del nivel de relevancia de un término**

Lucene proporciona el nivel de relevancia de los documentos recuperados en función de los términos de la consulta. Para aumentar el nivel de relevancia de un término se utiliza el símbolo '`^`' con un factor de incremento (un número) al final del término de búsqueda. Cuanto más alto sea ese factor más relevante será ese término en la búsqueda.

Esto permite controlar la relevancia de un documento aumentando el nivel de relevancia de sus términos. Por ejemplo, si se desea buscar

```
jakarta apache
```

y se desea que el término "`jakarta`" sea más relevante se utilizaría el símbolo ^ con un factor de aumento del nivel de relevancia al lado del término:

```
jakarta^4 apache
```

Con esto se consigue que los documentos en los que aparece el término `jakarta` resulten más relevantes para la búsqueda. Esta técnica también se puede utilizar con frases, no sólo con términos individuales:

```
"jakarta apache"^4 "jakarta lucene"
```

El factor de relevancia por defecto es 1. Debe ser un número positivo, pero puede ser menor que 1, (por ejemplo 0.2).

6.1.4 Operadores Booleanos

Los operadores Booleanos permiten combinar términos mediante operadores lógicos. Lucene admite los siguientes operadores Booleanos: AND, '`+`', OR, NOT y '`-`' (Nota: Los operadores Booleanos deben escribirse en mayúsculas).

- **OR**

El operador OR es el operador conjunción y es el operador Booleano por defecto. Esto es, si no aparece ningún operador Booleano entre dos términos de una consulta, se utiliza el operador OR. El operador OR actúa sobre dos términos y recupera un documento si alguno de los dos términos especificados aparece en el documento. Su comportamiento es equivalente a la unión de dos conjuntos. El símbolo `||` tiene el mismo significado y se puede usar en lugar de la palabra OR.

Para buscar documentos que contengan "`jakarta apache`" o sólo "`jakarta`" se utilizaría la siguiente consulta:

```
"jakarta apache" jakarta
```

o

```
"jakarta apache" OR jakarta
```

- **AND**

El operador AND recupera documentos en los aparezcan los dos términos de la consulta, en cualquier parte del texto del documento. Su comportamiento es equivalente a la intersección de conjuntos. Se puede utilizar el símbolo && en lugar de la palabra AND.

Para buscar documentos que contengan "`jakarta apache`" y "`jakarta lucene`" se utilizaría la consulta:

```
"jakarta apache" AND "jakarta lucene"
```

- **+**

El operador '+' exige que el término que aparece a continuación exista en alguno de los campos del documento.

Para buscar documentos que contengan el término "`jakarta`" y que puedan contener "`lucene`" la consulta sería:

```
+jakarta lucene
```

- **NOT**

El operador NOT excluye de la búsqueda los documentos que contienen el término que aparece a continuación del NOT. Este comportamiento es equivalente a la diferencia de conjuntos. El símbolo '`!`' puede ser utilizado en lugar de NOT.

Para buscar documentos que contienen "`jakarta apache`" pero no "`jakarta lucene`" habría que utilizar la consulta:

```
"jakarta apache" NOT "jakarta lucene"
```

Nota: El operador NOT no puede ser utilizado con un único término. Por ejemplo, la siguiente búsqueda no devolvería ningún resultado:

```
NOT "jakarta apache"
```

- **–**

El operador '`-`' excluye de la búsqueda los documentos que contienen el término que aparece después del símbolo '`-`'.

Para buscar documentos que contengan "`jakarta apache`" pero no "`jakarta lucene`" habría que utilizar la siguiente consulta:

```
"jakarta apache" - "jakarta lucene"
```

6.1.5 Agrupaciones

Lucene permite el uso de paréntesis para agrupar oraciones para formar subconsultas. Esto es muy útil para controlar la lógica Booleana de una consulta.

Para buscar "`jakarta`" o "`apache`" y "`web`" se usaría la consulta:

```
(jakarta OR apache) AND web
```

Esto evita confusiones y asegura que "`web`" debe existir y cualquiera de los dos términos ("`jakarta`" o "`apache`") pueden existir.

6.1.6 Agrupamiento por campo

Lucene permite el uso de paréntesis para agrupar varias expresiones de búsqueda para un único campo.

Para buscar un título que contenga la palabra "`retorno`" y la frase "`pantera rosa`" se utilizaría la consulta:

```
titulo:(+retorno +"pantera rosa")
```

6.1.7 Escapar caracteres especiales

Lucene permite escapar caracteres especiales que forman parte de la sintaxis de consulta. La lista de caracteres especiales es:

`+ - && || ! ( ) { } [ ] ^ " ~ * ? : \`

Para escapar estos caracteres se utiliza '`\`' antes del carácter. Por ejemplo para buscar`(1+1):2` se utilizaría la consulta:

```
\(1\+1\)\:2
```

6.2 USO DE LOS SCRIPTS IMPORT / EXPORT PARA BACKUP

Los scripts `import` y `export` se encuentran disponibles en el directorio `tools/arn-index` de la plataforma. Se ofrecen en dos versiones: `import.sh` y `export.sh` (para sistemas Linux) y `import.bat` y `export.bat` (para sistemas Windows).

El script `export` permite exportar todos los metadatos y la configuración de un servidor ARN-Indexer a un fichero zip. Los metadatos exportados son los mismos que cuando se utilizan las opciones equivalentes de la herramienta de administración (ver sección 4.3.1).

El formato de invocación del script es el siguiente:

```
export –h host –p port –l login -P password –f outputFilename
```

donde:
`-h host` indica el nombre o dirección IP de la máquina en la que está lanzado el servidor.
`-p port` indica el número de puerto en el que está lanzado el servidor.
`-l login` indica el nombre de usuario con el que se realizará la conexión al servidor.
`-P password` indica la contraseña con la que se realizará la conexión al servidor.
`-f outputFilename` indica el nombre del fichero zip al que se exportarán los metadatos.

A continuación, se muestra un ejemplo de ejecución del comando `export`:

```
export –h localhost –p 9000 -l admin -P admin –f backup.zip
```

Este comando exporta los metadatos completos del servidor ARN-Indexer que se está ejecutando en la máquina local en el puerto 9000. El acceso al servidor se realiza con el usuario `admin` con password `admin`. El resultado de la exportación se guarda en un fichero llamado `backup.zip`.

El script `import` permite importar los metadatos contenidos en un fichero zip obtenido mediante la utilidad `export` del servidor ARN-Indexer.

El formato de invocación del script es el siguiente:

```
import –h host –p port –l login -P password –f inputFilename [–replace]
```

donde:

`-h host` indica el nombre o dirección IP de la máquina en la que está lanzado el servidor.
`-p port` indica el número de puerto en el que está lanzado el servidor.
`-l login` indica el nombre de usuario con el que se realizará la conexión al servidor.
`-P password` indica la contraseña con la que se realizará la conexión al servidor.
`-f inputFilename` indica el nombre del fichero zip que contiene los metadatos a importar.
`-replace` es un argumento opcional que especifica si se desean reemplazar los elementos existentes por los incluidos en el fichero que se está importando, en el caso de que ya existan.

Por ejemplo:

```
import –h localhost –p 9000 -l admin -P admin –f backup.zip -replace
```

importa los metadatos contenidos en `backup.zip` en el servidor que se está ejecutando en la máquina local en el puerto 9000. El acceso al servidor se realiza con el usuario `admin` con password `admin`. Los mensajes de información y/o advertencia que devuelve el servidor como resultado de la importación se escriben por consola..

BIBLIOGRAFÍA

[DENINST] Guía de Instalación de la Plataforma Denodo . Denodo Technologies, 2008.

[LUCE] Apache Lucene, *http://lucene.apache.org/*

[LOG4J] Log4j, *http://logging.apache.org/log4j/docs/*

[MSEX] Microsoft Exchange Server. *http://www.microsoft.com/exchange/*

[SCHED] Guía del Administrador de Denodo Scheduler. Denodo Technologies 2008.

[SLF] Salesforce.com. On-demand Customer Relationship Management. *http://www.salesforce.com/*